UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/814,551	03/31/2004	Patrick R. Guido	RSW920030282US1/4541-016	7434

67419 7590 09/18/2008
COATS & BENNETT/IBM
1400 CRESCENT GREEN
SUITE 300
CARY, NC 27518

EXAMINER	
ORR, HENRY W	

ART UNIT	PAPER NUMBER
2176	

MAIL DATE	DELIVERY MODE
09/18/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/814,551
Filing Date: March 31, 2004
Appellant(s): GUIDO ET AL.

Guido et al.
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 7/14/2008 appealing from the Office action mailed 2/7/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,995,103 A	Ashe, Dylan B.	11-1999
5,920,313 A	Diedrichsen et al.	7-1999
5,694,561 A	Malamud et al.	12-1997
2005/0198585 A1	Haynes, Christopher M.	9-2005

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-5, 10-22 and 25-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ashe as cited above in view Diedrichsen et al. (hereinafter "Diedrichsen"), U.S. Patent No. 5, 920, 313.

Claim 1:

Ashe teaches a GUI window environment including a plurality of windows (see abstract, col. 5 lines 42-49). **(claim 1; i.e., providing a GUI environment including a plurality of windows;)**

Ashe teaches establishing a first affinity group comprising multiple related windows but less than all of said plurality of windows in a GUI environment (see col. 1 line 60 thru col. 2 line 63- teaches all the plurality of windows to be spread amongst multiple window layers, therefore each window layer ("affinity group") is comprised of multiple related windows (i.e., two or more) pertaining to a single application but less than all of said plurality of windows). **(claim 1; i.e., establishing, by a user, a first affinity group comprising a subset of two or more but less than all of said plurality of windows in said GUI environment)**

Ashe teaches said first affinity group (e.g. window layer) including windows associated with an application (see col. 1 lines 60-65).

Ashe fails to expressly teach said first affinity group including windows associated with at least two different applications.

However, Diedrichsen explicitly recites *"Different applications can also be organized into groups of applications, each of which are related by function"* (see col. 1 lines 63-65- a group can comprise of different applications). **(claim 1; i.e., said first**

affinity group including windows associated with at least two different applicants, such that the windows comprising said first affinity group are related;)

Therefore in the same field of endeavor of associating related user interface objects, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the group of windows within the window layer associated with a single application as taught by Ashe to be associated with different applications as taught by Diedrichsen to provide the benefit of easily identifying which objects are related to which other objects (see Diedrichsen; abstract, col. 2 lines 1-4, col. 3 lines 56-65).

Ashe teaches raising the z-order of a window layer (“affinity group") above other windows in other window layers when a window in the window layer is selected (see col. 2 lines 18-45). **(claim 1; i.e., and raising a z-order of windows in said first affinity group above other windows in said GUI environment when any one window in said first affinity group is selected.)**

Claim 2:

Ashe teaches raising all windows in a window layer to the front of the display screen (“top level z-order”) of said GUI environment (see col. 2 lines 2-45 teaches depending on the priority class of window layer, a window layer with the highest priority is capable of being raised to the top level z-order). **(claim 2; i.e., wherein raising the z-order of windows in said first affinity group above other windows in said GUI**

environment when any one window in said first affinity group is selected comprises raising all windows in said first affinity group to a top level z-order of said GUI environment.)

Claim 3:

Ashe fails to expressly teach tiling the windows in window layer (“affinity group”).

However, Diedrichsen teaches tiling the window objects belonging to a set membership such that said windows may simultaneously occupy the top level z-order of said GUI environment (see col. 7 lines 60-67 thru col. 8 lines 14, Figure 5C and 5D). **(claim 3; i.e., further comprising tiling the windows in said first affinity group such that said windows may simultaneously occupy the top level z-order of said GUI environment.)**

Therefore in the same field of endeavor of associating related user interface objects, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify windows belonging to an window layer (“affinity group”) as taught by Ashe to be tiled and result in the windows simultaneously occupying the top level z-order as taught by Diedrichsen to provide the benefit of easily identifying all members of the set (“affinity group”) to which a currently selected window object belongs (see Diedrichsen; abstract, col. 3 lines 56-65).

Claim 4:

Ashe teaches raising all other windows in window layer (“window layer”) to one or more z-order levels immediately below the top level (see col. 10 lines 24-59- teaches a window layer (“affinity group”) with modal and floating windows, in which floating windows (i.e., other windows) are raised to one z-order level (e.g. selected modal window”) below a modal window because modal windows always appear in front of floating windows within a window layer).

(claim 4; i.e., wherein raising the z-order of windows in said first affinity group above other windows in said GUI environment when any one window in said first affinity group is selected comprises raising the selected window to a top level z-order of said GUI environment, and raising all other windows in said first affinity group to one or more z-order levels immediately below the top level.)

Claim 5:

Ashe teaches designating an “affinity relationship” between existing windows in said GUI by the user (see col.11 lines 63-67- teaches organizing a selected set of existing windows into a window group). **(claim 5; i.e., wherein establishing said first affinity group of windows comprises designating an affinity relationship between existing windows in said GUI by the user.)**

Claim 10:

Ashe teaches creating new windows from existing windows, wherein the existing window and the new window have an affinity group relationship (see col. 9 lines 53-67 thru col. 10 lines 1-14). **(claim 10; i.e., wherein establishing said first affinity group of windows comprises creating one or more new windows from an existing window by the user, said existing window and said new windows having an affinity group relationship.)**

Claim 11:

Ashe teaches creating new windows by executing an affinity group window creation command; wherein the existing window and the new window have an affinity group relationship (see col. 9 lines 53-67 thru col. 10 lines 1-14). **(claim 11; i.e., wherein creating one or more new windows from an existing window by the user comprises: selecting an existing window; and creating a first new window by executing an affinity group window creation command; whereby said first new window created has an affinity group relationship with said existing window.)**

Examiner interprets application program ("existing window") creating a new window to be equivalent to executing an affinity group window creation command as recited in claim 11 because the application program can specify what order the new window will be displayed within an existing "affinity group relationship" amongst already defined windows belonging to an application program (see col. 9 lines 53-67 thru col. 10 lines 1-14).

Claim 12:

Ashe teaches creating new windows by executing an affinity group window creation command; wherein the existing window and the new window have an affinity group relationship (see col. 9 lines 53-67 thru col. 10 lines 1-14). **(claim 12; i.e., selecting either said existing window or said first new window; and creating a second new window by executing an affinity group window creation command; whereby said second new window created has an affinity group relationship with said existing window and said first new window.)**

Claim 13:

Ashe teaches a GUI window environment including virtual desktops (see col. 5 lines 42-49). **(claim 13; i.e., wherein said GUI environment includes virtual desktops.)**

Claim 14:

Claim 14 is a method claim and is substantially encompassed in method claim 1; therefore claim 14 is rejected under the same rationale as method claim 1 above. In addition to the rationale of claim 1, Ashe teaches creating multiple window layers ("first and second" affinity groups). Thus, Ashe is capable of switching between two or more affinity groups in a GUI environment as recited in claim 14 (see col. 1 line 60 thru col. 2 line 63).

Claim 15:

Ashe teaches a NewWinGroup method capable of organizing a “second affinity group” comprising all windows in said GUI environment not yet included in any affinity group (see col.11 lines 63-67).

Claim 16:

Claim 16 is a method claim and is substantially encompassed in method claim 2; therefore claim 16 is rejected under the same rationale as method claim 2 above.

Claim 17:

Claim 17 is a method claim and is substantially encompassed in method claim 3; therefore claim 17 is rejected under the same rationale as method claim 3 above.

Claim 18:

Claim 18 is a method claim and is substantially encompassed in method claim 4; therefore claim 18 is rejected under the same rationale as method claim 4 above.

Claim 19:

Claim 19 is a system claim and is substantially encompassed in method claim 1; therefore the system claim is rejected under the same rationale as method claim 1 above.

Claim 20:

Claim 20 is a system claim and is substantially encompassed in method claim 2; therefore the system claim is rejected under the same rationale as method claim 2 above.

Claim 21:

Claim 21 is a system claim and is substantially encompassed in method claim 3; therefore the system claim is rejected under the same rationale as method claim 3 above.

Claim 22:

Claim 22 is a system claim and is substantially encompassed in method claim 4; therefore the system claim is rejected under the same rationale as method claim 4 above.

Claim 25:

Claim 25 includes a program embodied on a computer readable medium to implement the steps that are substantially encompassed in method claim 1; therefore the claim is rejected under the same rationale as method claim 1 above.

Claim 26:

Claim 26 includes a program embodied on a computer readable medium to implement the steps that are substantially encompassed in method claim 2; therefore the claim is rejected under the same rationale as method claim 2 above.

Claim 27:

Claim 27 includes a program embodied on a computer readable medium to implement the steps that are substantially encompassed in method claim 3; therefore the claim is rejected under the same rationale as method claim 3 above.

Claim 28:

Claim 28 includes a program embodied on a computer readable medium to implement the steps that are substantially encompassed in method claim 4; therefore the claim is rejected under the same rationale as method claim 4 above.

7. **Claims 6-9, 23 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ashe in view of Diedrichsen as cited above, in further view of Malamud et al. (hereinafter “Malamud”), U.S. Patent No. 5,694,561.**

Claims 6 and 7:

Neither Ashe nor Diedrichsen expressly teach designating an affinity relationship using a drag and drop technique as recited in claims 6 and 7.

However, Malamud teaches designating a project group relationship (“affinity relationship”) between existing windows by the user comprising: selecting a first window; dragging said first window to an iconic display area of a project group folder (“affinity group icon on a second window”); and dropping said first window on said iconic display area of a project group folder, thereby establishing an project group relationship between said first window and project group folder (“second window”) (see col. 2 lines 30-55, col. 8 lines 59-67, col. 9 lines 1-8).

(claim 6; i.e., wherein designating an affinity relationship between existing windows by the user comprises: selecting a first window; dragging said first window to an affinity group icon on a second window; and dropping said first window on said affinity group icon of said second window, thereby establishing an affinity group relationship between said first and second window.)

Malamud teaches the capability to add a third window to the project group folder (“affinity group”) as recited in claim 7 (see col. 2 lines 30-55, col. 8 lines 59-67, col. 9 lines 1-8).

(claim 7; i.e., selecting a third window; dragging said third window to an affinity group icon on either said first or second window; and dropping said third window on said affinity group icon of said first or second window, thereby adding said third window to said affinity group.)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the grouping of windows to established an affinity relationship as taught by Ashe to include a drag and drop technique as taught by

Malamud to provide the benefit of easily forming the desired windows into a related group (see Malamud; col. 1 lines 33-57, Figure 2).

Claims 8 and 9:

Neither Ashe nor Diedrichsen expressly teach designating an affinity relationship using a keystroke combination technique as recited in claims 8 and 9.

Claims 8 and 9 are method claims and are substantially encompassed in method claims 6 and 7 respectively; therefore the method claims 8 and 9 are rejected under the same rationale as method claims 6 and 7 above.

In addition to the rationale of claims 6 and 7, one of ordinary skill at the time of the invention was made would have been able to implement the drag and drop technique as taught by Malamud as a keystroke combination to perform the limitations as recited in claims 8 and 9 (see Malamud; col. 9 lines 9-15).

Claim 23:

Claim 23 is a system claim and is substantially encompassed in method claim 6; therefore the system claim is rejected under the same rationale as method claim 6 above.

Claim 24:

Claim 24 is a system claim and is substantially encompassed in method claim 8; therefore the system claim is rejected under the same rationale as method claim 8 above.

8. **Claims 6-9, 23 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ashe in view of Diedrichsen as cited above, in further view of Haynes (hereinafter "Haynes"), U.S. Published Application No. 2005/0198585 A1.**

Claims 6 and 7:

Neither Ashe nor Diedrichsen expressly teach designating an affinity relationship using a drag and drop technique as recited in claims 6 and 7.

However, Haynes teaches designating a relationship using a drag and drop technique as cited in claims 6 and 7 (see par. 19, claims 13 and 14, Figure 3). **(claim 6; i.e., wherein designating an affinity relationship between existing windows by the user comprises: selecting a first window; dragging said first window to an affinity group icon on a second window; and dropping said first window on said affinity group icon of said second window, thereby establishing an affinity group relationship between said first and second window.)**

(claim 7; i.e., selecting a third window; dragging said third window to an affinity group icon on either said first or second window; and dropping said third window on said affinity group icon of said first or second window, thereby adding said third window to said affinity group.)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the grouping of windows to established an affinity relationship as taught by Ashe to include a drag and drop technique as taught by Haynes to provide the benefit of easily forming the desired windows into a related group (see Haynes; par. 19, Figure 3).

Claims 8 and 9:

Neither Ashe nor Diedrichsen expressly teach designating an affinity relationship using a keystroke combination technique as recited in claims 8 and 9.

However, Haynes teaches designating a relationship using a keystroke combination technique as cited in claims 8 and 9 (see par. 19, claims 15 and 16, Figure 3). **(claim 8; i.e., wherein designating an affinity relationship between existing windows by the user comprises: selecting a first window; executing a first keystroke combination in said first window; selecting a second window; and executing a second keystroke combination in said second window, thereby establishing an affinity group relationship between said first and second window.)**

(claim 9; i.e., selecting a third window; executing said first keystroke combination in said third window; selecting either said first or second window; and executing said second keystroke combination in said selected first or second window, thereby adding said third window to said affinity group.)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the grouping of windows to established an affinity relationship as taught by Ashe to include a keystroke combination technique as taught by Haynes to provide the benefit of easily forming the desired windows into a related group (see Haynes; par. 19, Figure 3).

Claim 23:

Claim 23 is a system claim and is substantially encompassed in method claim 6; therefore the system claim is rejected under the same rationale as method claim 6 above.

Claim 24:

Claim 24 is a system claim and is substantially encompassed in method claim 8; therefore the system claim is rejected under the same rationale as method claim 8 above.

(10) Response to Argument

35 U.S.C. § 103 Rejections Under Ashe and Diedrichsen

The claims recite grouping different and independent applications.

Appellant argues that every independent claim recites grouping GUI windows running different and independent applications and those applications are different in that they are not the same application, and are independent in that they are not logically or functionally related.

Examiner first notes that the terms “different" and "independent" are not expressly defined in the instant specification.

Therefore, Examiner interprets the terms “different" and “independent” as would one of ordinary skill in the art at the time the invention was made.

Appellant uses the example of grouping windows associated with a word processor application, an email client, and web browser and alleges that the applications are different and independent because the applications are not the same and **<u>are not logically or functionally related</u>**. (emphasis added).

Examiner agrees that the Appellant’s interpretation of the term “different” would be reasonable to one of ordinary skill in the art.

In respect to the Appellant’s interpretation of the term “independent” interpretation, Examiner notes that paragraph 14 in the instant specification recites " the Internet browser in window 36, the e-mail client in window 38, and the word processor in window 40 are **<u>all related to the same task or operation</u>**--namely, writing a report”. (emphasis added)

In addition, in paragraph 21, the instant specification recites "affinity groups are useful for managing windows 36, 38, 40 **<u>running related applications</u>**”. (emphasis added)

Since the instant specification mentions the different applications **<u>to be related</u>**, it appears that the instant specification contradicts the Appellant's interpretation of the term "independent" being **<u>not logically or functionally related</u>** because the different applications as described in the instant specification are at least **<u>related by the function</u>** of writing a report. (emphasis added).

<u>Ashe discloses grouping windows associated with the same application.</u>

Appellant argues that Ashe fails to teach or suggest an affinity group of GUI windows and manipulating the z-order of the group, where the windows are associated with at least two *different* applications.

Examiner notes to Appellant that the Examiner does not rely solely on Ashe to teach or suggest an affinity group of GUI windows and manipulating the z-order of the group, where the windows are associated with at least two *different* applications. In fact, Examiner submits that such limitation would have been obvious over Ashe in view of Diedrichsen.

Ashe teaches a window layer ("affinity group") of GUI windows and manipulating the z-order of the layer, where the windows are associated with a single application (see col. 1 lines 60-67, col. 2 lines 1-45).

Diedrichsen teaches different applications to be organized into a group ("affinity group")(see col. 1 lines 60-65).

Therefore in the same field of endeavor of associating related user interface objects, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the group of windows within the window layer associated with a single application as taught by Ashe to be associated with different applications as taught by Diedrichsen to provide the benefit of easily identifying which objects are related to which other objects (see Diedrichsen; abstract, col. 2 lines 1-4, col. 3 lines 56-65).

Ashe does not disclose grouping windows associated with different applications.

Appellant argues that Ashe fails to teach or suggest grouping windows associated with different and independent applications, as recited in claims 1, 14, 19 and 25.

As explained above, Examiner does not rely solely on Ashe to teach or suggest grouping windows associated with different and independent applications. Examiner relies on Ashe in view of Diedrichsen to teach such features as explained above.

Appellant further argues that Ashe’s background discussion of z-order does not teach grouping less than all application windows for z-order manipulation with respect to other GUI windows.

Examiner respectfully disagrees.

Ashe teaches multiple window layers, wherein each layer is a set of all windows associated with a single application (see col. 1 lines 60-65). Therefore, when there are multiple applications, there must be multiple layers. Ashe further teaches selecting a window within a window layer causing that window layer to overlie other window layers (see col. 2 lines 32-45). Therefore, multiple window layers represent all application windows and an individual window layer represents a group of windows less than all application windows. Thus, Ashe's background discussion of z-order does teach or suggests grouping less than all application windows for z-order manipulation with respect to other GUI windows.

Diedrichsen discloses grouping windows related as parent/child.

Appellant argues that Diedrichsen does not disclose grouping windows, for simultaneous z-order manipulation, that are associated with independent applications – only applications that "are related to the selected window" as parent/child.

Examiner notes to Appellant that the Examiner does not rely solely on Diedrichsen to teach or suggest grouping windows, for simultaneous z-order manipulation, that are associated with independent applications. In fact, Examiner submits that such limitation would have been obvious over Ashe in view of Diedrichsen.

Ashe teaches grouping windows within a window layer for simultaneous z-order manipulation, that are associated with a single application (see col. 1 lines 60-67, col. 2 lines 1-45).

Diedrichsen teaches different applications to be organized into a group (“affinity group”)(see col. 1 lines 60-65). Based on the instant specification examples of different applications being “independent”, Examiner submits that one of ordinary skill in the art at the time the invention was made would have interpreted the different applications as taught by Diedrichsen to be "independent".

Therefore in the same field of endeavor of associating related user interface objects, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the group of windows within the window layer associated with a single application as taught by Ashe to be associated with different (“independent”) applications as taught by Diedrichsen to provide the benefit of easily identifying which objects are related to which other objects (see Diedrichsen; abstract, col. 2 lines 1-4, col. 3 lines 56-65).

Diedrichsen does not disclose grouping independent windows.

Appellant mentions that the claims recite grouping independent applications and that **related applications are not independent**. (emphasis added).

As noted above, paragraph 14 in the instant specification recites "the Internet browser in window 36, the e-mail client in window 38, and the word processor in window 40 are **all related to the same task or operation**--namely, writing a report". (emphasis added)

In addition, in paragraph 21, the instant specification recites "affinity groups are useful for managing windows 36, 38, 40 **running related applications**". (emphasis added)

Since the instant specification mentions the different applications **to be related**, it appears that the instant specification contradicts the Appellant's interpretation of the term "independent" being **not related applications** because the different applications as described in the instant specification are at least **related by the function** of writing a report. (emphasis added). Thus, this discrepancy raises the question as whether the Appellant intended to add new subject matter to claims because the different applications as described by the instant application **are related**.

Based on the instant specification examples of different applications being "independent", Examiner submits that one of ordinary skill in the art at the time the invention was made would have interpreted the different applications as taught by Diedrichsen to be "independent".

Appellant further argues that Diedrichson offer no suggestion of grouping windows associated with different and independent applications for z-order manipulation, and contains no teaching of any mechanism for doing so.

As stated above, Examiner notes to Appellant that the Examiner does not rely solely on Diedrichsen to teach or suggest grouping windows associated with different and independent applications for z-order manipulation. Examiner submits that such limitation would have been obvious over Ashe in view of Diedrichsen as explained above.

The claimed invention is not obvious over Ashe and Diedrichsen.

Appellant argues that neither Ashe nor Diedrichsen, separately or in combination, fairly teach or suggest grouping together two or more, but less than all, windows in a GUI environment, the windows associated with different, independent applications, for simultaneous z-order manipulation of the windows in the group with respect to all other windows in the GUI environment.

Examiner respectfully disagrees.

Ashe teaches multiple window layers, wherein each layer is a set of all windows associated with a single application (see col. 1 lines 60-65). Therefore, when there are multiple applications, there must be multiple layers. Ashe further teaches selecting a window within a window layer causing that window layer to overlie other window layers (see col. 2 lines 32-45). Therefore, multiple window layers represent all application windows and an individual window layer represents a group of windows less than all application windows. Thus, Ashe's background discussion of z-order does teach or suggests grouping less than all application windows for z-order manipulation with respect to other GUI windows.

Ashe teaches grouping windows within a window layer for simultaneous z-order manipulation, that are associated with a single application (see col. 1 lines 60-67, col. 2 lines 1-45).

Diedrichsen teaches different applications to be organized into a group ("affinity group")(see col. 1 lines 60-65). Based on the instant specification examples of different applications being "independent", Examiner submits that one of ordinary skill in the art at the time the invention was made would have interpreted the different applications as taught by Diedrichsen to be "independent".

Therefore in the same field of endeavor of associating related user interface objects, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the group of windows within the window layer associated with a single application as taught by Ashe to be associated with different ("independent") applications as taught by Diedrichsen to provide the benefit of easily identifying which objects are related to which other objects (see Diedrichsen; abstract, col. 2 lines 1-4, col. 3 lines 56-65).

Based on the rationale set forth above, Examiner submits grouping together two or more, but less than all, windows in a GUI environment, the windows associated with different, independent applications, for simultaneous z-order manipulation of the windows in the group with respect to all other windows in the GUI environment is obvious over Ashe in view of Diedrichsen.

35 U.S.C. § 103 Rejections Under Ashe, Diedrichsen, and Malamud

Malamud does not cure the failure to establish a prima facie case of obviousness.

Appellant arguments with respect to Malamud not disclosing simultaneous z-order manipulation of groups has been addressed by Examiner with the rationale set forth under Ashe in view of Diedrichsen. Therefore, in response to the Malamud arguments see the above remarks.

35 U.S.C. § 103 Rejections Under Ashe, Diedrichsen, and Haynes

Haynes does not cure the failure to establish a prima facie case of obviousness.

Appellant arguments with respect to Haynes not disclosing simultaneous z-order manipulation of the window among which focus is cycled has been addressed by Examiner with the rationale set forth under Ashe in view of Diedrichsen. Therefore, in response to the Haynes arguments see the above remarks.

Conclusion

For the above reasons, it is believed that the rejections should be sustained.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

Respectfully submitted,

/Henry Orr/

Conferees:

/Rachna S Desai/
Primary Examiner, Art Unit 2176

/Doug Hutton/
Doug Hutton
Supervisory Primary Examiner
Technology Center 2100